31st Annual INCOSE international symposium

virtual event

July 17 - 22, 2021

Security as a Functional Requirement

By Dr. Keith D. Willett (FuSE Systems Security Project)

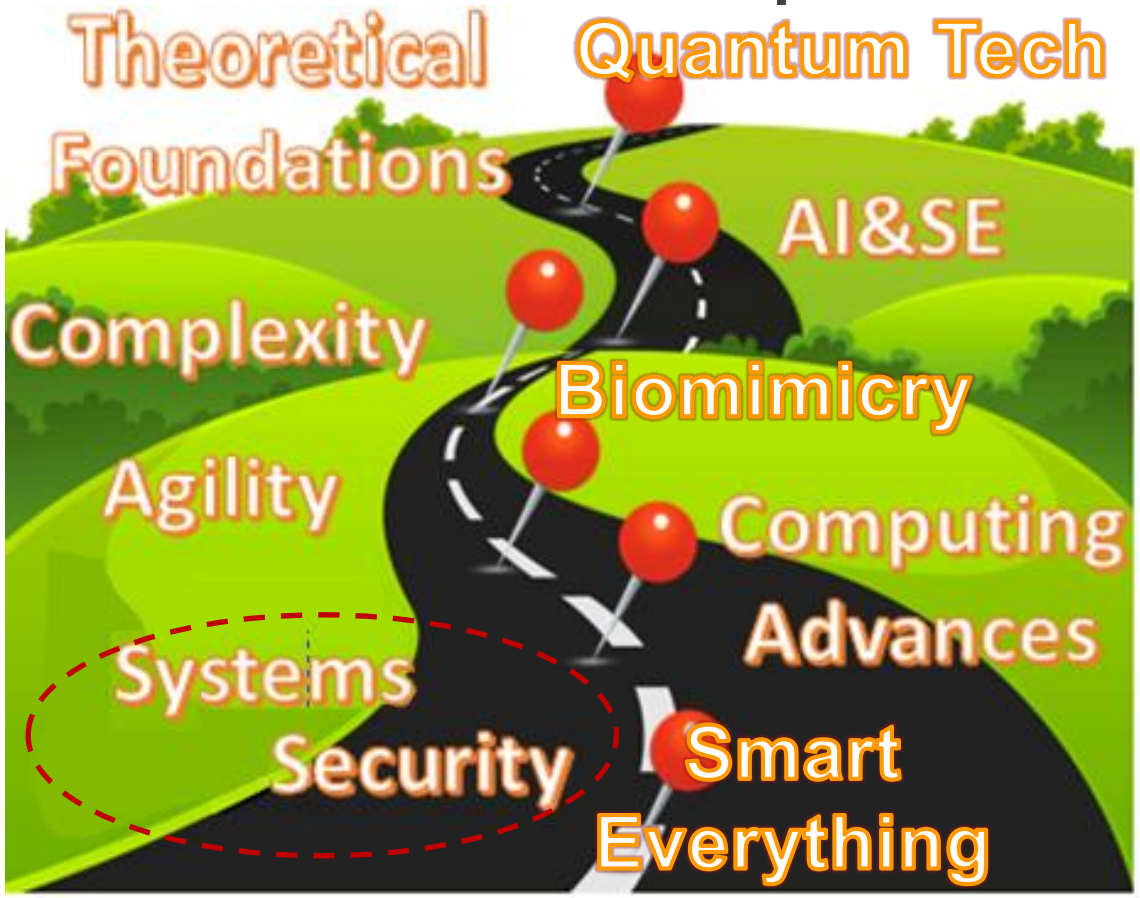
www.incose.org/symp2021

INCOSE FuSE

Future of Systems Engineering

FuSE Collaborative Community

FuSE Road Map



FuSE System Security Charter 2020

Systems Security in the Future of Systems Engineering
(a FuSE initiative topic project)

Lead: Rick Dove. Team:
US DoD: Keith Willett
ISSS: Delia Pembrey MacNamara
NDIA: Holly Dunlap, Corey Ocker
SERC: Tom McDermott.

What will good look like when we use FuSE to deliver systems?

1. Security agility is in practice.
2. System and component behavior is monitored for anomalous operation.
3. Systems are built for trust.
4. System components are self protective.
5. Security is Embedded in Systems.
6. All stakeholders share common security vision and respect.

What will good look like in 2023-2025?

1. Security responsibility and expertise is integrated in the SE-team.
2. Security is viewed as a functional requirement.
3. Security agility will have some effective working patterns in practice as an early base line.
4. Strategies for shared security vision and respect in early practice.

What will good look like by end of 2020?

1. Multi-organization collaboration is active.
2. Initial foundation concepts for FuSE Security identified.
3. Projects to develop and publish some of the foundation concepts are active.

What is stopping us from doing this now?

1. SE relates to SSE as an independent specialty practice.
2. Security is viewed as a non-functional cost and ROI value is difficult to verify.
3. Security standards compliance is considered sufficient.
4. Actionable research is in early stages.
5. Contracts and projects detail features and requirements up front rather than desired capabilities that allow innovative solutions.

Action Plan

1. IS20 initial foundation papers:
   Techno-Social Contracts for Security Orchestration.
   Contextually Aware Agile Security.
   Architecting the Future of System Security.
2. Mid 2020: Periodic web workshops in process identifying additional foundation areas.
3. Ongoing: Recruit foundation developers.
4. Late 2020: Additional foundation papers in process.

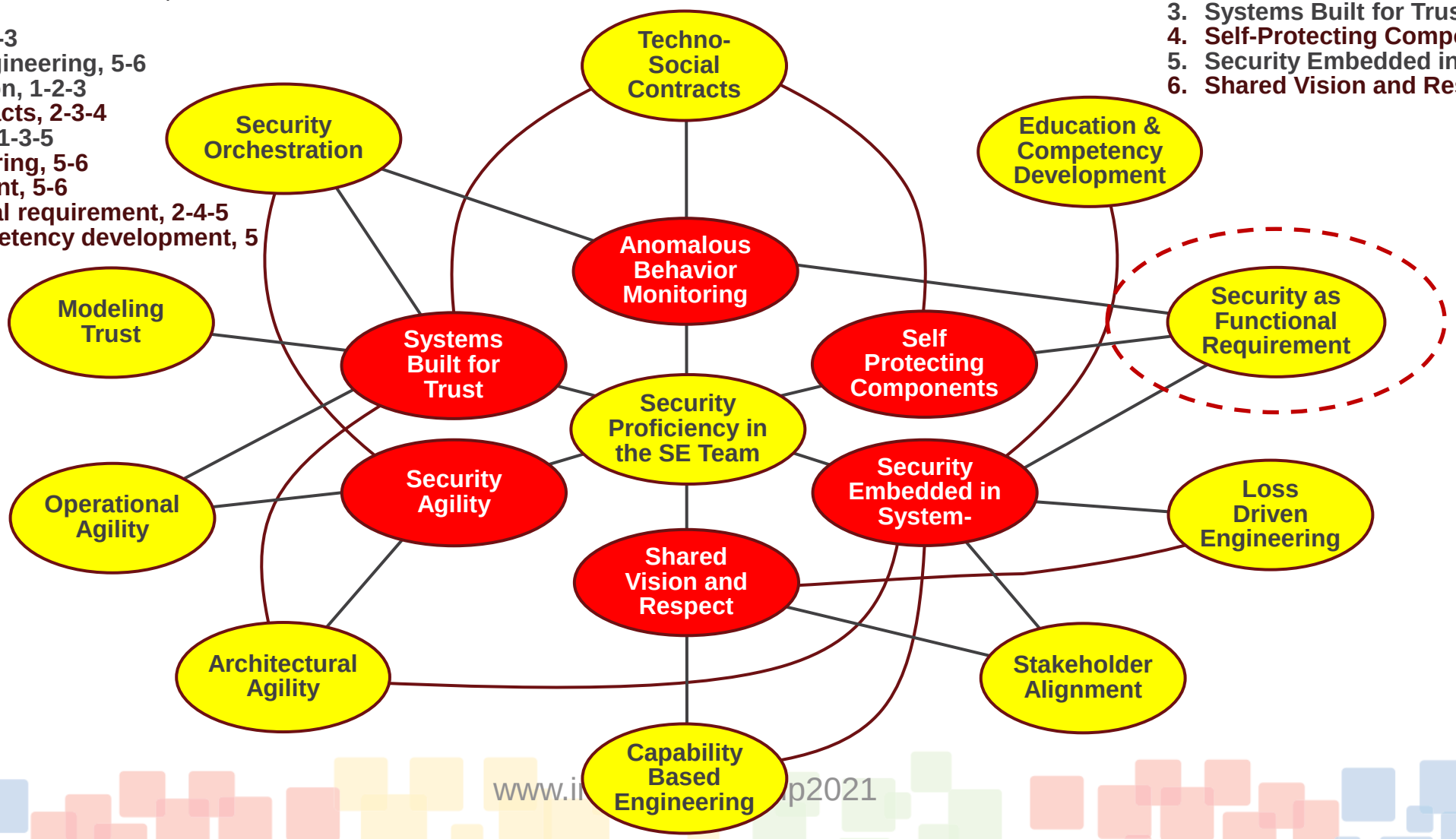
Activity Map

Concepts to Objectives

1. Security proficiency in the SE team, 1-2-3-4-5-6
2. Modeling trust, 3
3. Operational agility, 1-3
4. Capability-based engineering, 5-6
5. Security orchestration, 1-2-3
6. Techno-social contracts, 2-3-4
7. Architectural agility, 1-3-5
8. Loss driven engineering, 5-6
9. Stakeholder alignment, 5-6
10. Security as functional requirement, 2-4-5
11. Education and competency development, 5

Objectives

1. Security Agility
2. Anomalous Behavior Monitoring
3. Systems Built for Trust
4. Self-Protecting Components
5. Security Embedded in System
6. Shared Vision and Respect



Security's Lament

"They call me 'non-functional' and try to make me feel good by labeling me a *quality attribute*. I'm just a forgotten -ility… tolerated, not wanted… an expense, not investment… a constraint, not an enabler… but I'm better than that, and I'm going to prove it! I *am* quality, and more…

I'm *functional*!"

Security as FR: Topic Synopsis

Problem	As a non-functional requirement, systems security does not get SE prime attention.
Need	SE responsibility for the security of systems.
Barriers	SE practice codifies security as a non-functional requirement.
Intent	Establish security as a functional requirement; inherently raise importance.
Value	Integrate security throughout the SE lifecycle processes.
Metrics	Presence of effective functional security requirements.
Notions	Common Criteria. Open Security Architecture. OMG Unified Architecture Framework. Industrial Internet of Things Security Framework.

Functional Requirement

- Qualitative description
  - Activity to perform (behavior)
  - Purpose to achieve (goal)

Security does stuff

Security accomplishes stuff

Goals

- Primary goal of any system
  - Value-delivery
- Some systems are **expendable**
  - Sustain value-delivery under *nominal* conditions
  - If something goes wrong, let it go
- Some systems are **protected**
  - Sustain value-delivery under *adverse* conditions
  - If something goes wrong, keep it going

Risk: Key Driver

- *Stakeholder needs* process
  - Adversity imposes potential loss
  - Determine the degree of acceptable loss
    - Stakeholder risk tolerance
  - Risk tolerance drives *need* for security

Discerning Stakeholder Needs (Design)

- **Risk Tolerance***
  - Formally express capacity to endure loss
- **Risk Posture***
  - Intentionally assumed position to address all risk
    - Accept, share, transfer, or mitigate

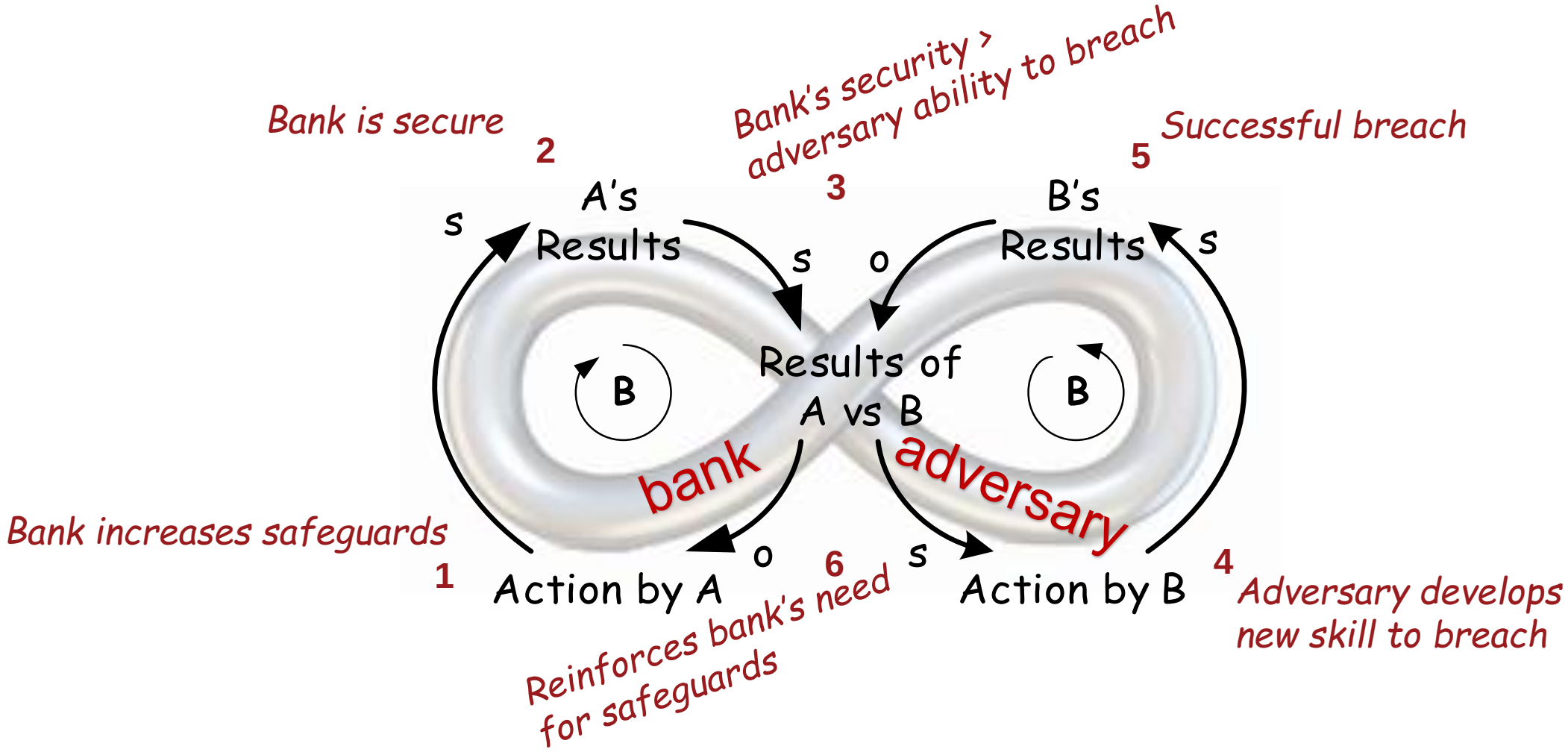
* Formal artifact

Ensuring Customer Needs (Operations)

- **Desired Security Posture***
  - Intentionally assumed position to enforce Risk Posture
- **Continual Monitoring Plan***
  - **Actual Security Posture** (trigger-based snapshots)
    - Compare to Desired Security Posture
    - Perform/generate **Gap Analysis***
  - **Gap Closure Plan***
    - Resource constraints; may take minutes to years

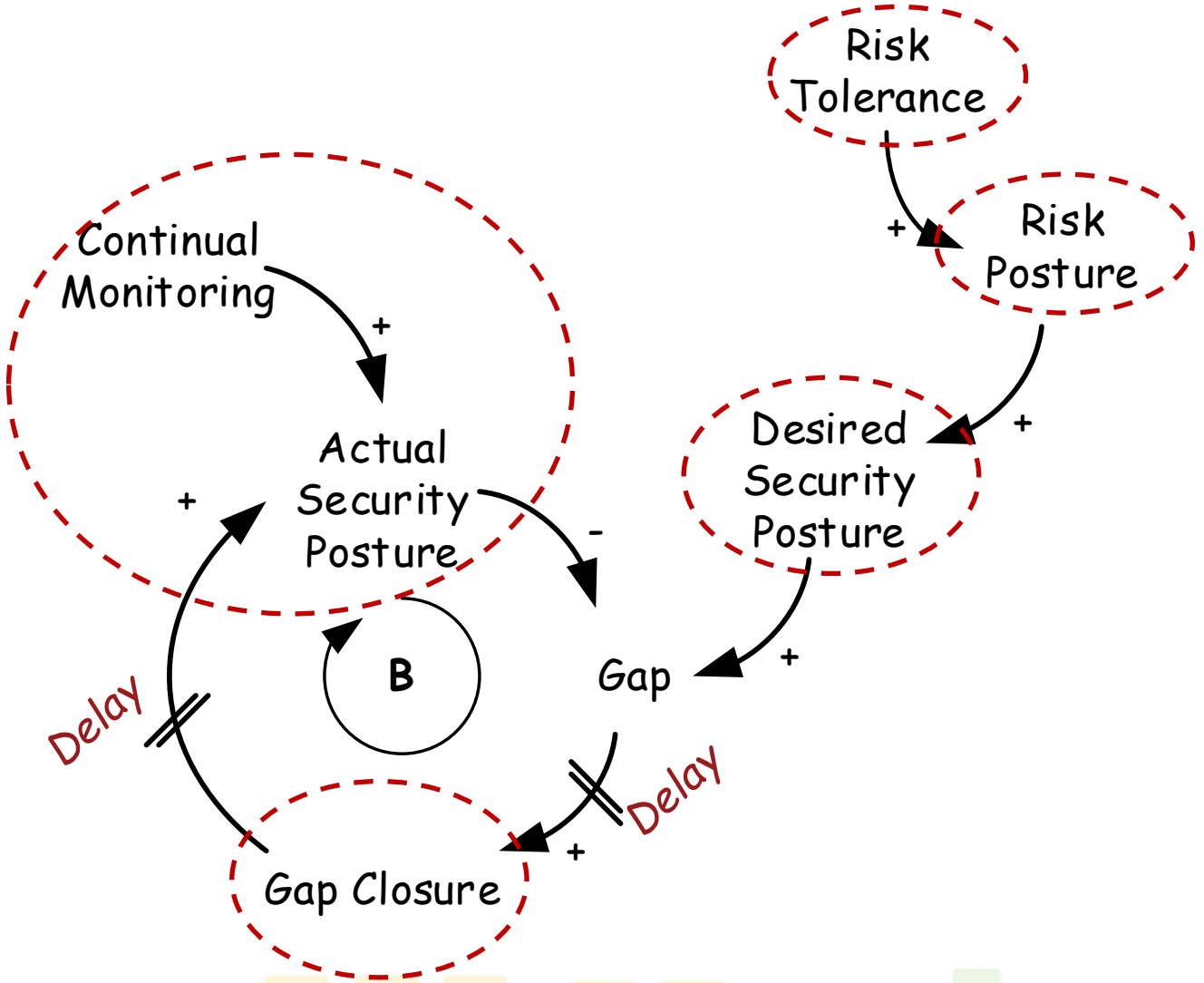
* Formal artifact

Security is an Infinite Game (Escalation Archetype)

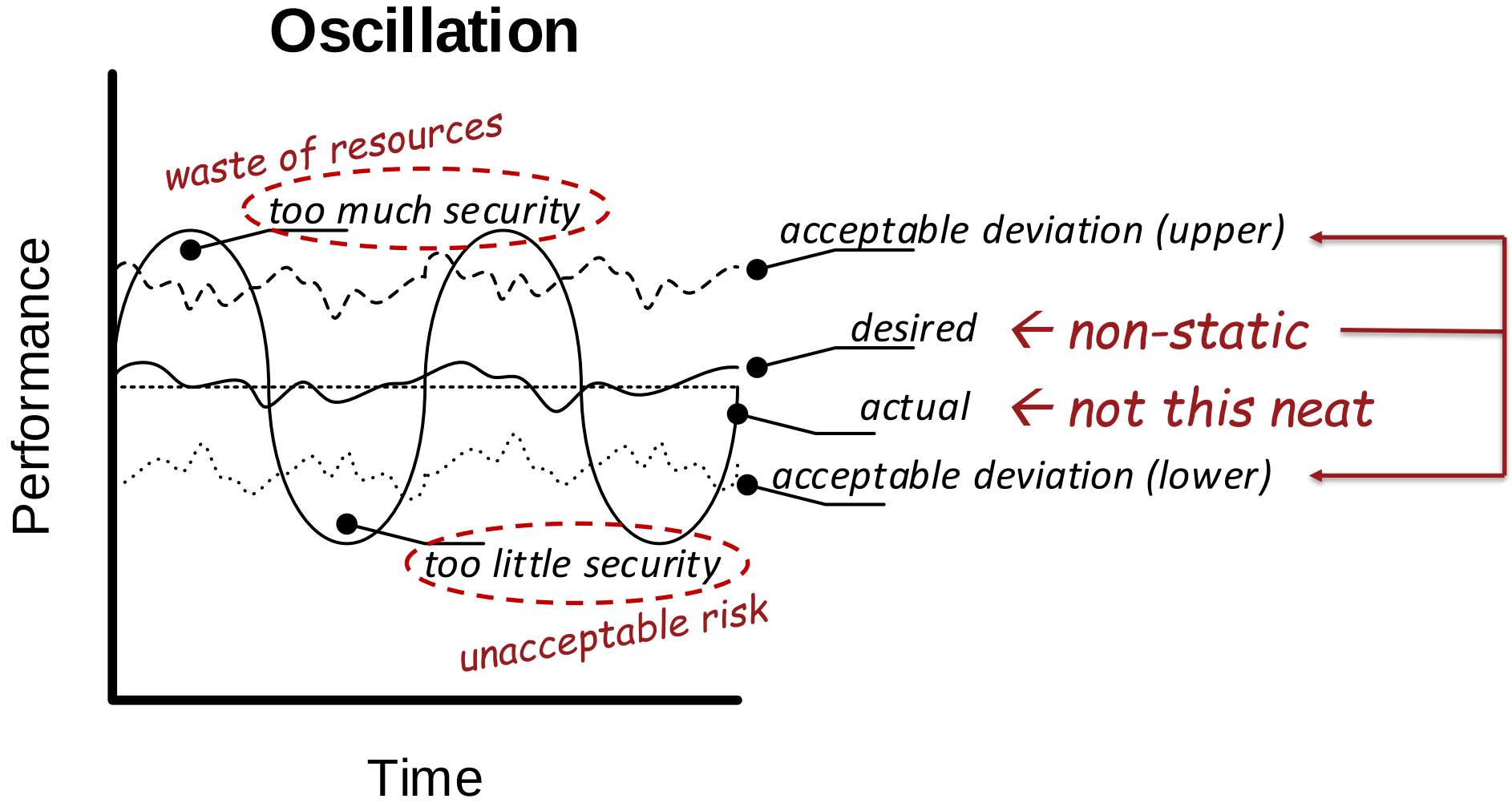


S = same; positive; + O = opposite; negative; -

Security Dynamics Oscillate

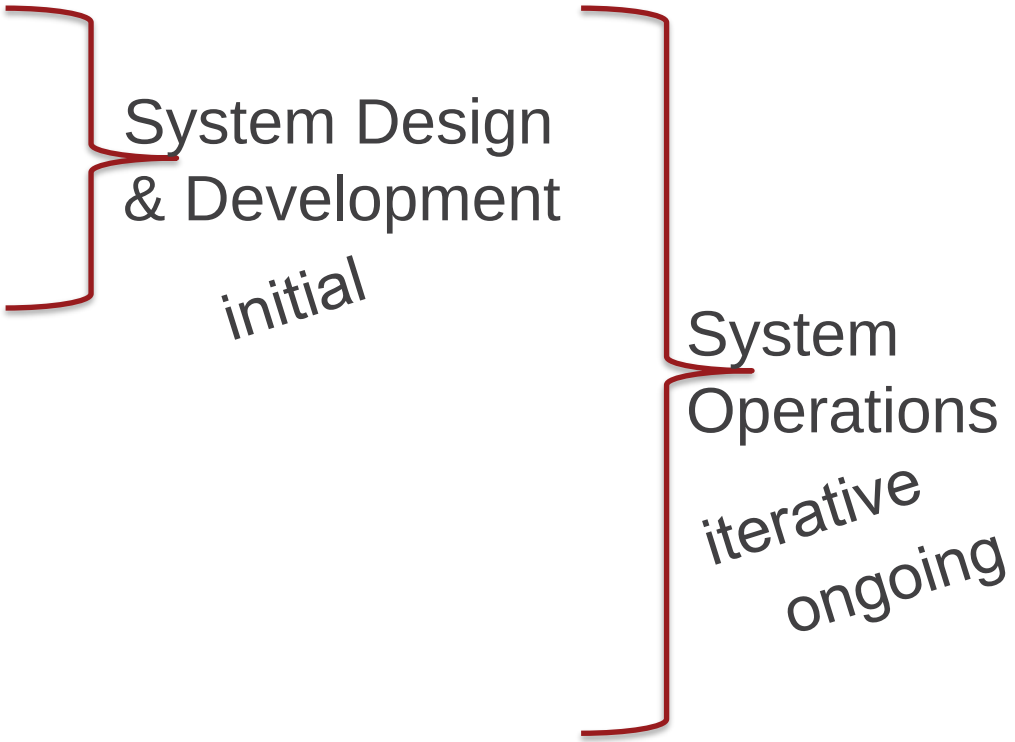


Security Dynamics Oscillation Graph



Additional Artifacts to SE Process

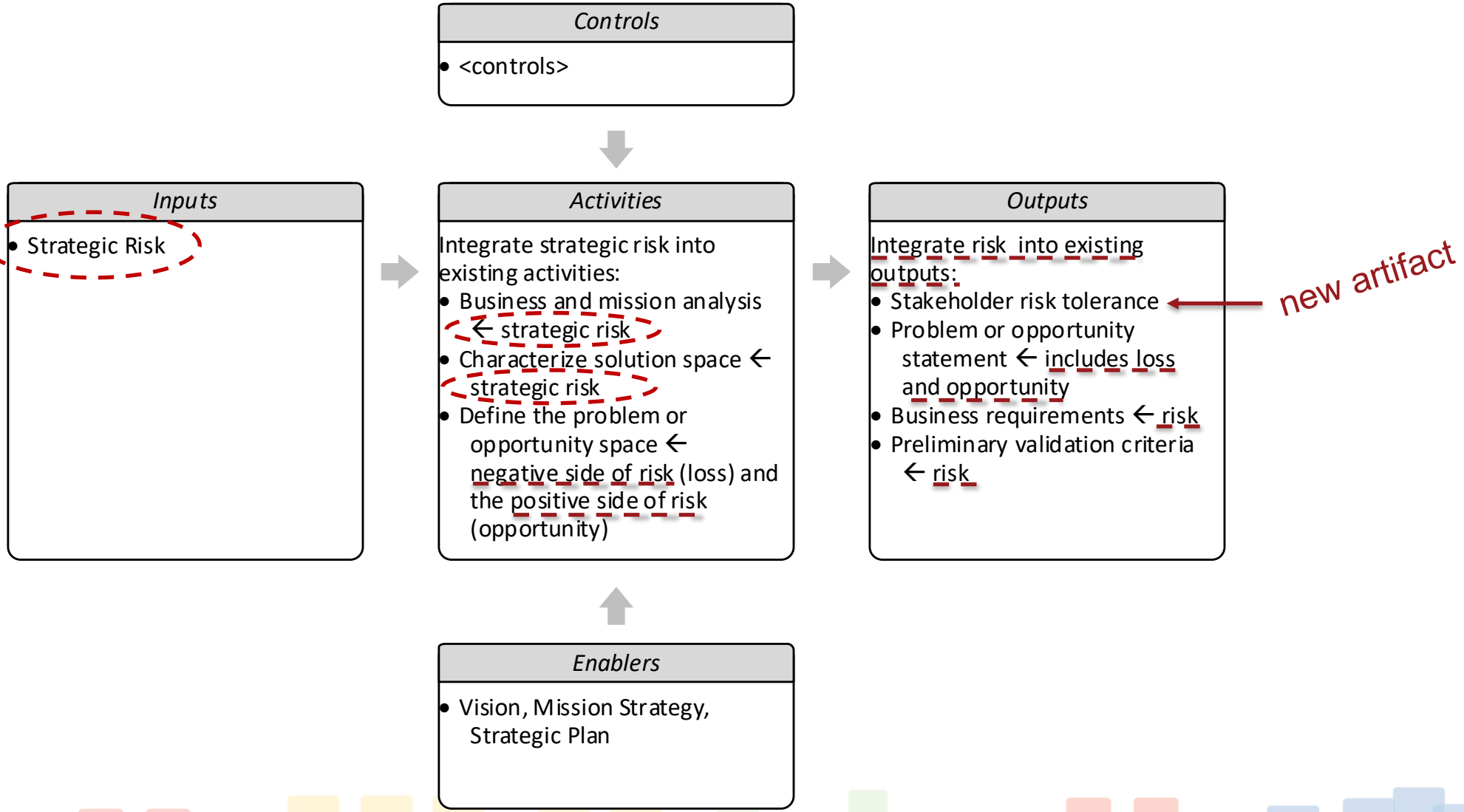
- Risk Tolerance
- Risk Posture
- Desired Security Posture
- Continual Monitoring Plan
- Actual Security Posture
- Gap Analysis
- Gap Closure Plan



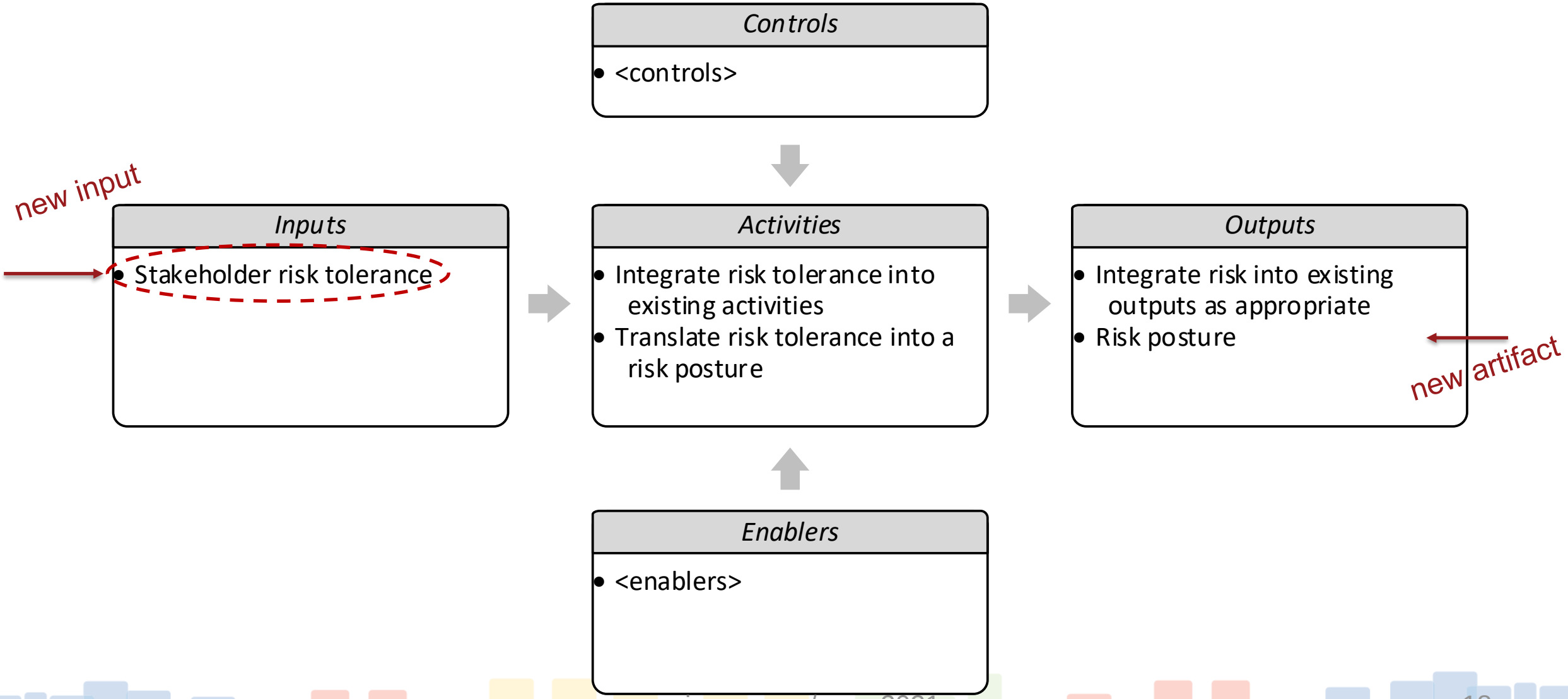
Integrate Security in SE Process

- Introduce security early in SE process
  - Subsequent processes absorb security
  - Security inherently part of system lifecycle

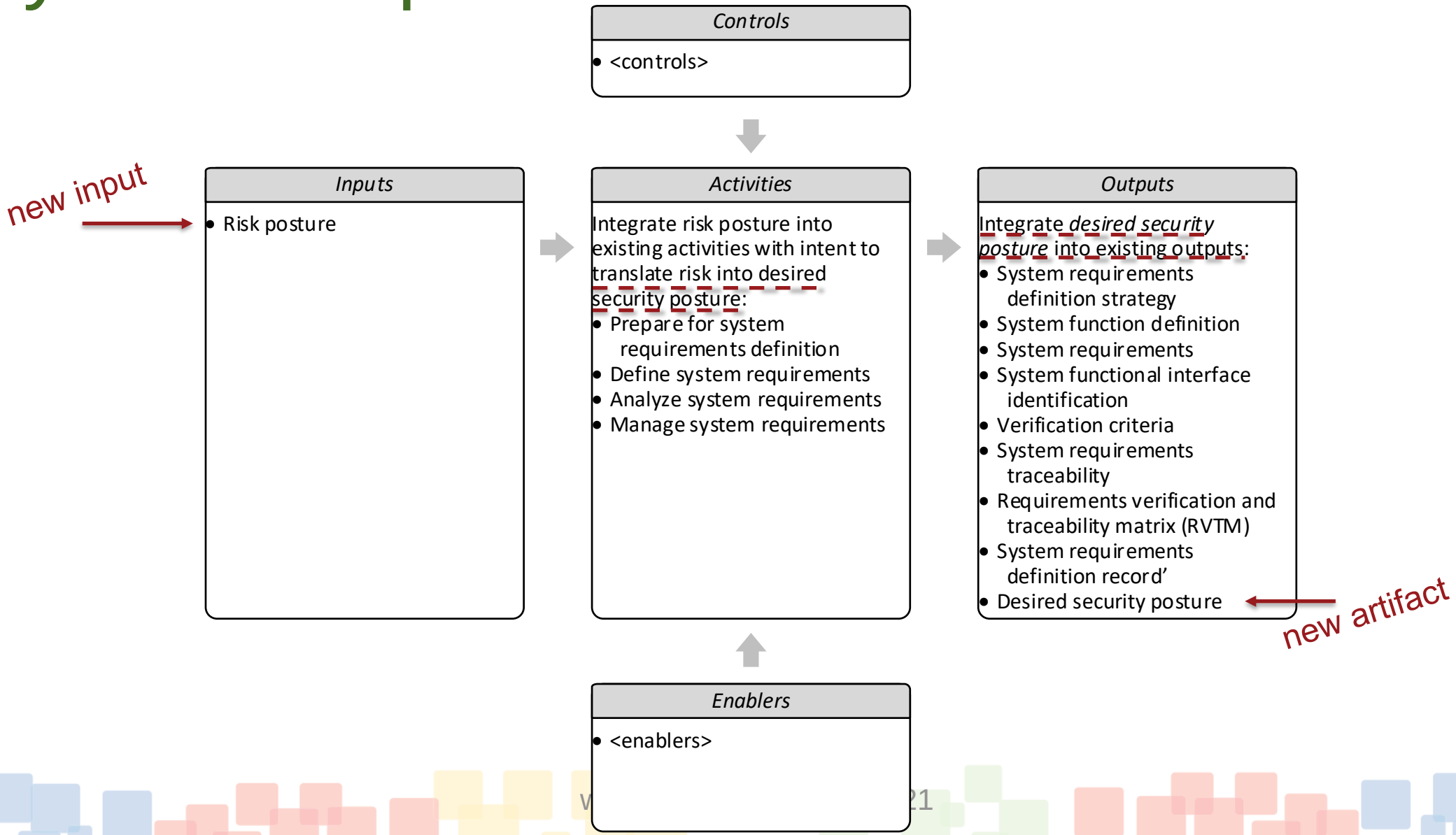
Business and Mission Analysis IPO



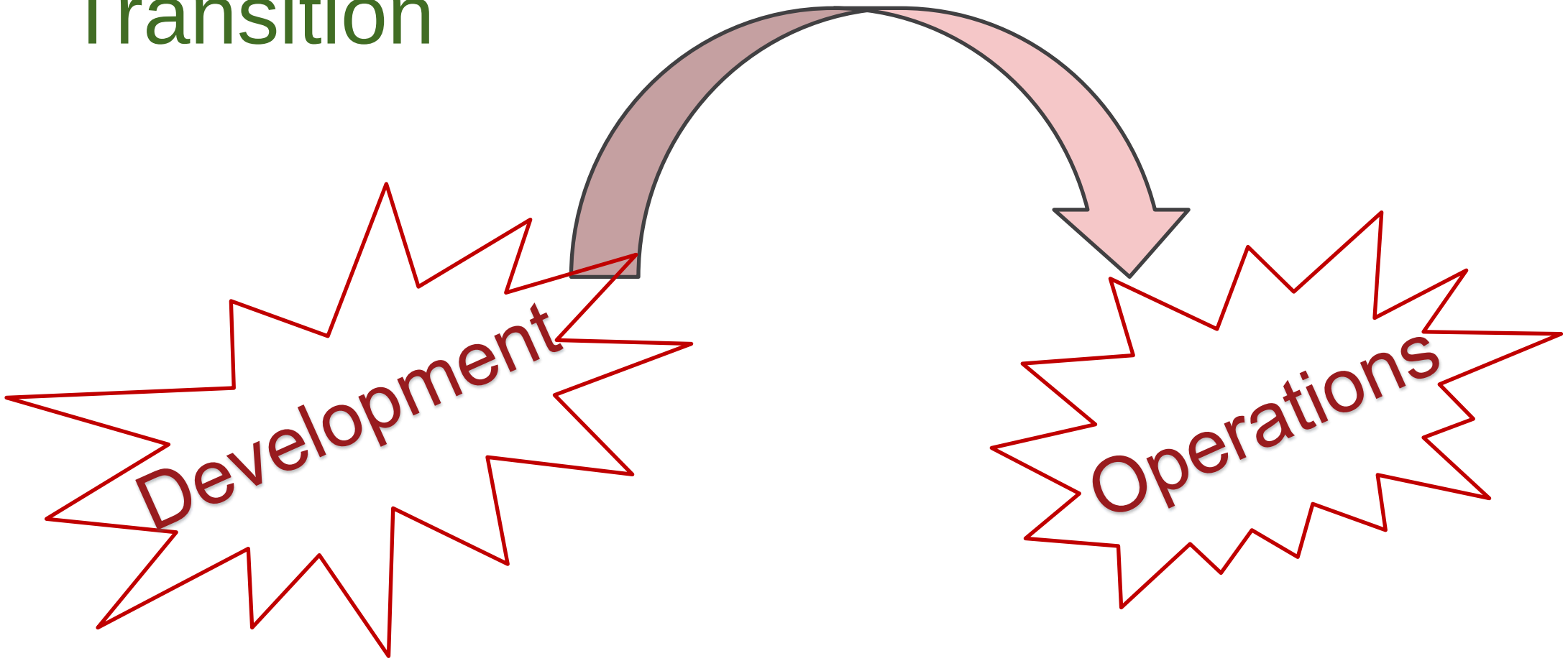
Stakeholder Needs & Rqmts Def IPO



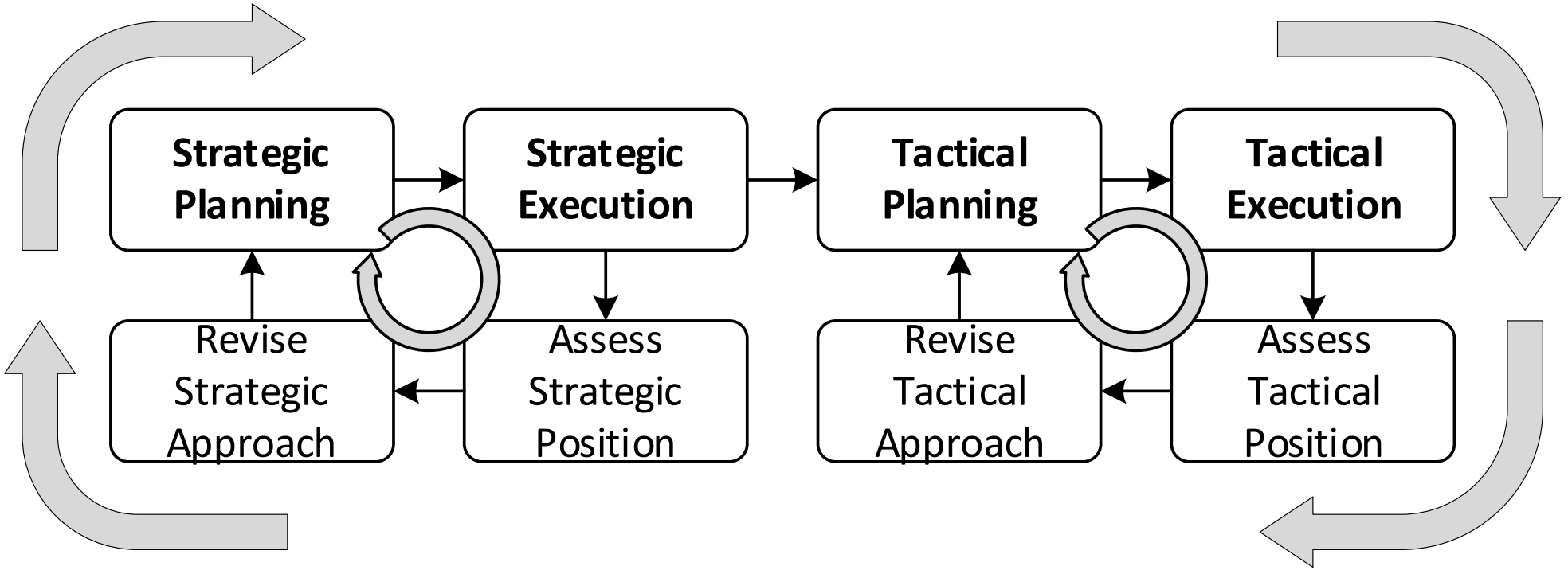
System Requirements Definition IPO



Transition



Setting Up Operational Agility



Manage Security as an Infinite Game

- ***Goals*** (value-delivery)… non-static… risk tolerance and risk posture are goals for security.
- ***Strategies*** support goals
  - *Function-driven* (what the system does to provide value-delivery), *loss-driven* (negative side of risk), and *opportunity-driven* (positive side of risk)… security is one domain under loss driven.
- ***Objectives*** are measurable steps within strategies.
  - May fluctuate as strategies adjust to new goals. Desired Security Posture captures security objectives.
- ***Methods*** are tactics, techniques, and procedures to achieve objectives.
  - Methods invoke solutions.
  - Continual monitoring is method to sustain acceptable degree of security.
- ***Solutions*** include safeguards (features, functions, tools, services)
- ***Measures*** are *X in regard to Y* for some aspect of the system or safeguard
  - X ∈ (binary, degree, statistics, probability, time, distance, quantity, accuracy, etc.)
  - Y ∈ (structure, state, behavior, function, functional exchange, contents, resources, environment, value-delivery, etc.)
  - Provide insight to degree of performance and/or achieving objective

Measurement Structure

Goal	Strategy	Objective	Method	Solution	Measure
<goal>	<supporting strategy>	<measurable step within strategy>	<TTPs to achieve objectives>	<tools; products, services>	<status, state>

Context Matters

- Expression of meaning and value
  - Social/cultural (who)
  - Technical (what)
  - Spatial (where)
  - Temporal (when)
  - Behavior (how)
  - Desire (why)

- Context frames need and risk tolerance
  - No universal expression

Example Goal Statements

- Optimize value-delivery: produce desired results
- Optimize risk (risk neutral)
- Minimize risk (risk averse)
- Maximize risk (risk seeking)
- Minimize threats
- Minimize threat efficacy
- Minimize vulnerabilities
- Minimize vulnerability efficacy
- Minimize negative impact
- Minimize negative effect
- Minimize negative consequences
- Maximize safeguard efficacy
- Maximize positive impact
- Maximize positive effect
- Maximize positive consequences
- Maximize learning (knowledge)
- Maximize profit
- Minimize cost
- … etc.

Example Needs-Value Statements

- Need **confidentiality** to safeguard our secrets.
- Need **integrity** to keep the system whole.
- Need **availability** to ensure the system is ready for use.
- Need **possession** to retain physical control; minimize effects of loss or theft.
- Need **authenticity** to ensure the system is compatible with reality; anti-deception.
- Need **utility** to ensure the system is fit for purpose.
- Need **non-repudiation** to ensure attribution; ensure anti-anonymity.
- Need **privacy** to ensure ability to remain unobserved and ability to be forgotten; ensure anonymity.
- Need **authorized use** to ensure resource control; minimize theft of service or misallocation of resources.
- Need **accountability** to ensure explain-ability; learn and minimize repeating mistakes; minimize malicious anonymity.
- Need **recoverability** to ensure ability to return to a desired state.
- … etc.

Example Functional Requirements

- The system shall [maintain | access] a set of user roles.
- The system shall [maintain | access] a set of allowable actions per role.
- The system shall [maintain | access] a set of unique identifications per person or non-person entity attempting to access the system.
- The system shall interoperate with the existing identity management infrastructure.
  - Assumption: the identity management system maintains roles and responsibilities (allowable actions per role).
- The system shall authenticate a claim of identity.
- The system shall authorize a claim of privilege.
- The system shall interoperate with the existing privilege management infrastructure.
- The system shall encrypt X; X ∈ (data at rest, data in transit, data in use)
- The system shall interoperate with the existing encryption key management infrastructure.
- The system shall support explicit blocking of X (blacklist).
- The system shall support explicit allowing of X (whitelist).
- The system shall safeguard data.
  - The system shall disclose data to only authorized users.
  - The system shall allow data modification only by authorized users.
  - The system shall clear memory upon process termination to remove data in use.
- The system shall provide an Internet homing beacon; *phone home* on a periodic basis.
- The system shall log X [logons | failed logons | Y folder access | Y file access].
- The system shall notify in real-time of X where X is any state, function, functional exchange, etc. outside of specified acceptable parameters / thresholds.

Supporting Infrastructure (Assumptions)

- **Identity management**
  - Establish, operate, and maintain identities and identity credentials
- **Privilege management**
  - Establish, operate, and maintain privileges and privilege credentials
    - Explicit allow, explicit deny
    - Default allow, default deny
    - Deny all unless explicitly allowed (don't trust until given a reason to trust)
    - Allow all unless explicitly denied (trust until given a reason not to trust)
    - Block all and allow only explicit (whitelist)
    - Allow all and block only explicit (blacklist)
- **Encryption key management**
  - Establish, operate, and maintain keys
  - Key storage and retrieval (recovery)
- This relates in part to the principle of *utility* where an encryption key may be lost and the data otherwise not usable without key recovery
- **Backup management**
  - Establish, operate, and maintain system backups; recoverability.
  - This relates to disaster recovery, continuity of operations, and to ransomware where wiping a system and restoring may be cheaper than paying ransom.
- **Physical security**
  - Establish, operate, and maintain perimeter safeguards including a subset of campus, building, floor, room, office, and workstation.
- … etc.

Example Performance Requirements

- The system shall authenticate users within X milliseconds with a deviation of no more than +/- Y milliseconds.
- The system shall verify actions against a blacklist and return a block decision within X milliseconds with a deviation of no more than +/- Y milliseconds.
- The system shall verify inputs against a whitelist and return a block decision within X milliseconds with a deviation of no more than +/- Y milliseconds.
- The system shall provide an option to log all X activities.
- The system shall maintain a log of all X activities in a moving window of Y [hours | days].
- … etc.

Loss-Driven Systems Engineering (LDSE)

- INCOSE INSIGHT Vol 23 Issue 4 December 2020
- Security is but one discipline that addresses loss
- Others:
  - Safety
  - Agility
  - Resilience
  - Sustainability
  - Survivability
  - Etc.
- Prompts additional requirements aligned to value-delivery under adversity

Conclusion

- Without security, system viability and relevance are:
  - Left to chance in a nominal world
  - Open to malicious attack in an adverse world
- Risk tolerance → security need → security requirements → integrated into system FRs
- Only system requirements
  - Part of which covers security
    - Aligns to need for sustaining value-delivery under adversity
  - No separate security requirements

Next Steps

- Revise SE practices
  - Integrate LDSE that includes security
  - Absorb LDSE as standard part of engineered systems
  - Stakeholder needs:
    - Provide value-delivery
    - Sustain value-delivery under *nominal* conditions
      - Security may not be a consideration for expendable systems
      - Principle: *conscious omission vs omission by oversight*
    - Sustain value-delivery under *adverse* conditions
      - Security ceases to be a separate consideration
      - There is no successful system without security

Questions
? ? ? ? ?

Backup/Reference Slides

FuSE Systems Security (SE) Approach

- Produce set of foundational topics:
  - Provide new and useful value to the state of practice.
  - Relevance to SE considerations.
  - Value proposition is in SE terms.
  - Supported by referenceable examples and/or knowledge base.
  - Doesn’t have sufficient published exposure for SE consideration.
  - Could be prototyped/employed now.
  - Has sufficient ecosystem/infrastructure to support application.
- Goal: inspire and instigate pursuit in the SE community.

31st Annual INCOSE international symposium

virtual event

July 17 - 22, 2021

www.incose.org/symp2021